

Spis treści

Przedmowa Michała Michalczuka	3
Przedmowa Bartosza Cytrowskiego	7
1. Wstęp: Po co TypeScript?	11
1.1 Czym jest TypeScript?	11
1.2 Statyczne typowanie	12
1.3 Inferencja typów	13
1.4 Dokumentacja i kontrakty	14
1.5 Pewność i niepewność	15
1.6 Pomoc narzędzi	16
1.7 Zalety TypeScripta – podsumowanie	17
1.8 Zanim przejdziesz dalej	18
1.9 Jak poprawnie odmieniać słowo TypeScript?	19
1.10 Konwencje używane w książce	20
2. TypeScript w 10 minut	23
2.1 Pierwsze linijki TypeScripta	23
2.2 TypeScript a JavaScript	24
2.3 Instalacja kompilatora	24
2.4 Kompilacja pliku	25
2.5 Dodajmy typy	25
2.6 Strict	27
2.7 Piaskownica	28
2.8 Edytor	28
2.9 Przykładowy projekt	29

2.10	Node.js	30
2.11	tsconfig.json.	31
2.12	Podsumowanie	32
3.	Wprowadzenie do przydatnych elementów z ES6+	33
3.1	ECMAScript	33
3.2	Klasy.	34
3.3	Moduły	35
3.4	Funkcje strzałkowe	36
3.5	let i const	37
3.6	Destrukturyzacja i skrócony zapis obiektów.	41
3.7	Domyślna wartość parametrów.	44
3.8	Rest	46
3.9	Spread.	46
3.10	Pozostałe nowości	47
3.11	Niestandardowe elementy języka.	48
3.12	Operator import(...)	50
3.13	Prywatne pola w klasach ze znacznikiem #.	50
4.	Typowanie statyczne i typowanie silne	51
4.1	Systemy typów	51
4.2	Typowanie dynamiczne.	52
4.3	Typowanie statyczne	53
4.4	Słabe typowanie	54
4.5	Typowanie silne	55
4.6	Typy silne, słabe, statyczne, dynamiczne...	56
4.7	ts-ignore.	57
4.8	Typowanie strukturalne, nominalne i duck typing.	57
4.9	Podsumowanie	61
5.	Typy elementarne	63
5.1	Podstawy	63
5.2	Słowo kluczowe type	74
5.3	Podsumowanie	75
6.	Funkcje	77
6.1	Argumenty funkcji.	77

6.2	Typ zwracany	77
6.3	Wyrażenia funkcyjne	78
6.4	Typ funkcji.	79
6.5	Inferencja typu argumentów	79
6.6	Parametry opcjonalne.	80
6.7	Parametry domyślne.	81
6.8	Funkcje wariadyczne	81
6.9	Przeładowywanie funkcji.	82
6.10	this.	85
6.11	Podsumowanie	86
7.	Klasy i interfejsy	87
7.1	Klasa.	87
7.2	Modyfikatory public, private i protected.	91
7.3	Klasa abstrakcyjna.	95
7.4	Interfejs.	95
7.5	Typ statyczny i typ instancji w klasach	103
7.6	Pola prywatne ES.	105
8.	Typy generyczne	109
8.1	Funkcje generyczne	109
8.2	Inferencja w generykach	110
8.3	Generyczne typy	111
8.4	Inne generyki	112
8.5	Ograniczenia generyków	113
8.6	Generyki wielu typów	113
8.7	Podsumowanie	115
9.	Inferencja typów i const	117
9.1	Przykład	117
9.2	Wnioskowanie typów argumentów	118
9.3	Typ wspólny	118
9.4	Inferencja czasem zawodzi.	119
9.5	Inferencja kontekstowa.	121
9.6	Cementowanie typów	122
9.7	Inferencja przy const i let	124
9.8	Podsumowanie	124

10. Kompatybilność typów	127
10.1 Kompatybilność: podtyp a przypisywanie	127
10.2 Kompatybilność strukturalna	128
10.3 Klasy z polami publicznymi	128
10.4 Klasy z polami prywatnymi	130
10.5 Kompatybilność podtypów	131
10.6 Przypisywanie literałów obiektów	132
10.7 Kompatybilność funkcji wariadycznych	133
10.8 Kompatybilność argumentów funkcji	134
10.9 Kompatybilność metod w obiektach	135
10.10 Kompatybilność typu zwracanego przez funkcje	136
10.11 Argumenty opcjonalne i rest.	137
10.12 Typy kowariantne, kontrawariantne, biwariantne i in- variantne	139
10.13 Powtórzenie	141
10.14 Więcej o kowariancji i kontrawariancji	142
10.15 Dla dociekliwych	144
10.16 Kowariancja i kontrawariancja przez analogię	145
10.17 Wariancja a mutowalność	147
10.18 Inferencja typów a wariancja	148
10.19 bivarianceHack	150
10.20 Dłaczego metody są biwariantne	151
 11. Enumy	 155
11.1 Enumy numeryczne	155
11.2 Enumy z ciągami znaków	159
11.3 Różnice pomiędzy enumami.	159
11.4 Enumy typowane nominalnie	161
11.5 Test wyczerpania.	161
11.6 Test wyczerpania z liczbami.	162
11.7 Test wyczerpania bez noImplicitReturns.	163
11.8 Kompatybilność obiektów i enumów	164
11.9 const enum.	165
11.10 Enum a literał stringa	166
11.11 Podsumowanie	166

12. Typy zaawansowane	167
12.1 <code>unique symbol</code>	167
12.2 Typ i interfejs	169
12.3 Łączenie deklaracji.	170
12.4 Aliasy typów.	171
12.5 Część wspólna i suma typów (unia)	172
12.6 <i>Index signature</i>	176
12.7 <i>Literal type</i>	178
12.8 <code>as const</code> – niemutowalne typy danych	179
12.9 Type guards	180
12.10 Pobieranie typu wartości	187
12.11 Przeładowywanie funkcji literałami	189
12.12 Typy rekurencyjne	190
12.13 Algebraiczne typy danych	192
12.14 Wyłuskiwanie typu	195
12.15 Mapped types	196
12.16 Tuple wariadyczne	202
13. Typy warunkowe	205
13.1 Co to są typy warunkowe?	205
13.2 Przykładowe użycie	206
13.3 Typy warunkowe na unii	206
13.4 Zagnieżdżanie.	207
13.5 Warunkowe typy dystrybtywne	208
13.6 Przykład użycia.	209
13.7 Opóźnione warunki	213
13.8 Kompatybilność typów warunkowych	214
13.9 <code>infer</code>	215
13.10 Podsumowanie	219
14. Unie w praktyce	221
14.1 Problem.	221
14.2 Podejście pierwsze	222
14.3 Podejście drugie.	225
14.4 Podejście trzecie	227
14.5 Co tu się stało?	230

14.6	Finalizacja	231
14.7	Podsumowanie	232
14.8	A rzutowanie?	232
15.	Bezpieczna praca z danymi	235
15.1	Dlaczego typy znikają po kompilacji.	236
15.2	Dane z zewnątrz i rzutowanie.	236
15.3	Generowanie typów z JSON	237
15.4	Walidacja	238
15.5	Zła walidacja.	240
15.6	Kiedy walidować	240
15.7	Biblioteki do walidacji.	241
15.8	io-ts i zod	242
15.9	Unie w praktyce.	245
15.10	Podsumowanie	247
16.	Typy nominalne	249
16.1	Czy typy pomagają?	250
16.2	Problemy z typowaniem strukturalnym	251
16.3	Typy nominalne w TS	253
16.4	Zestawienie rozwiązań	261
16.5	Biblioteki	263
16.6	Typy nominalne a walidacja w io-ts	264
16.7	Podsumowanie	265
17.	Własna walidacja, typy warunkowe i testowanie typów	267
17.1	Tworzenie własnego walidatora	267
17.2	Przechowywanie informacji o typach	268
17.3	Walidator z konfiguracją	270
17.4	Walidacja obiektów.	271
17.5	Nasz własny walidator	273
17.6	Testowanie typów	274
17.7	dtslint	274
17.8	Podsumowanie	276
18.	Implementacja systemu jednostek w TS	277
18.1	Koncepcja	277

18.2	Liczby Peana	278
18.3	Implementacja	281
18.4	Biblioteki	284
18.5	Podsumowanie	285
19.	Migracja z JavaScriptu	287
19.1	Konfiguracja dla łagodnego przejścia	287
19.2	Migracja w dół lub w górę	289
19.3	Sprawdzanie poprawności plików JS i JSDoc	290
19.4	Podsumowanie	294
20.	Pliki definicji typów .d.ts	295
20.1	Repozytorium DefinitelyTyped	295
20.2	Dodawanie typów do istniejących bibliotek w JS	296
20.3	Tworzenie bibliotek w TypeScriptie	299
20.4	Podsumowanie	300
21.	Popularne problemy z TS	303
21.1	Poprawne typowanie <code>Array#filter</code>	303
21.2	Typ <code>Object.keys()</code> jest niepoprawny	307
21.3	Wszystkie elementy tablicy są zdefiniowane	309
21.4	Funkcja przyjmuje obiekty, które mają więcej pól, niż powinny	311
21.5	Otagowane unie nie zawsze działają poprawnie	313
A.	Dodatek: Przydatne typy	315
A.1	Typy wbudowane w TypeScript	315
A.2	Inne przydatne typy	324
A.3	<code>ts-essentials</code> i <code>ts-toolbelt</code>	328